**Project Documentation**

Dong Lei

UALR

The database was built to accommodate the specific requirements of an online retailer. There are several tables in the database, each performing a distinct function.

User account data is kept in the first table, Login. A one-to-one correspondence exists between the account number (Acc No) and the customer ID (Customer ID). Each user's password is kept in this database, along with a log of their activity. For safety and identification reasons, this table is essential.

Each customer's details are saved in the Customer table. The customer's name, physical and virtual addresses, and email addresses, as well as their gender, are all part of this data set. The information in this table is used for customer relationship management and to monitor user behavior on the website.

The Bill's table is where we track what our customers have bought. Payment IDs (payment IDs) are linked to their respective Product IDs (product IDs) and Customer IDs (customer IDs) in order to track individual purchases. This database additionally keeps track of the payment type (PaymentType). This table is essential for keeping tabs on sales and stock levels.

Data about the company's inventory may be found in the Product table. The Product ID assigned to each item is used to associate it with client orders. This database also keeps track of the product's name (ProductName) and the date it was made (ManufactureDate). In order to keep tabs on stock and sales, this table is essential.

Many queries are provided in the application database to facilitate data extraction from the tables. Information on sales, purchases, and stock levels are all relevant to these inquiries. Information such as the number of sales for each product, the total amount for each payment option, and the total number of payments made by each client may all be retrieved from the database with a few simple queries. These kinds of inquiries may be utilized to understand company performance and as a basis for future choices.

In summary, the application's database is made to accommodate the requirements of an online retailer. The database allows the company to keep tabs on sales, manage inventories, and provide top-notch customer service by keeping information on user accounts, customer data, purchases, and items (Monger, Baron & Lu, 2009). The database may be used to automate processes, enforce business standards, and get insights into operational efficiency with its powerful set of queries and database objects.

**Conceptual Design**

The database's conceptual design is the abstract model of the data and its interconnections. It's a simplified model that emphasizes the system's data needs and limitations (Nuijten & Barel, 2023).

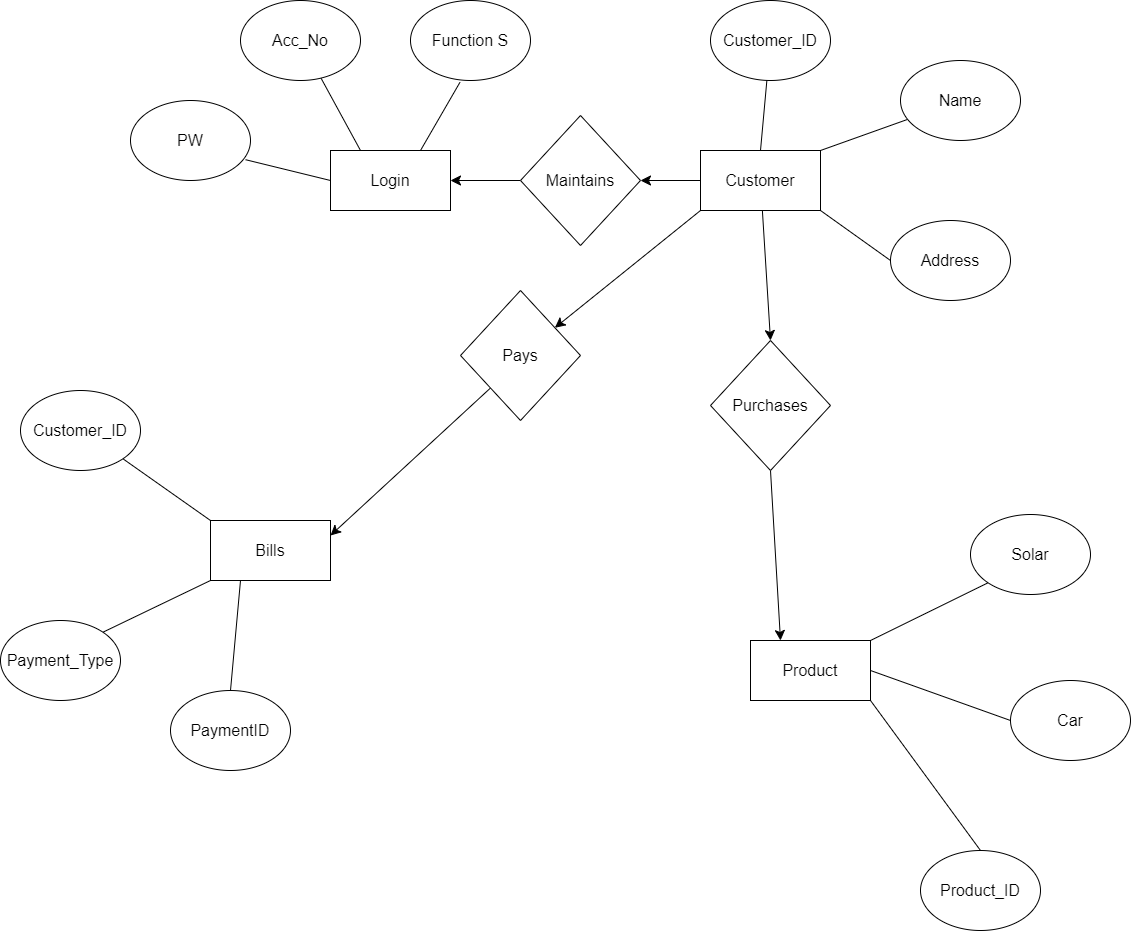
The entities and their connections form the basis of the conceptual design of the supplied database (Sciore, 2020). Client, Login, Products, and Bills are the primary entities. In this context, an entity stands in for a data collection that is useful to the program.

Customers' personal details, such as name, address, email, gender, and a unique identifier (Customer ID), are stored in an object called "Customer." The Login entity stores information about a client's login credentials, such as their account number (Acc No), customer ID, password, and account activity. The Product entity stores information on the goods produced by the business, such as the product's name, Product ID, and date of production. Bills is an object that stores details about invoices given to clients by the business, such as the payment ID, customer ID, product ID, and payment type.

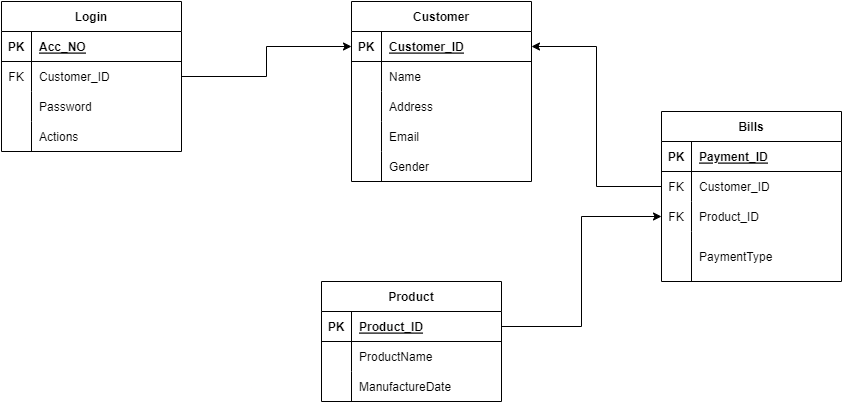
The connections between the elements are also crucial to the overall structure of the Schema. In this scenario, the connections between the entities are many. Each client has one and only one login, as shown by the foreign key reference between the Login and Customer entities. Each bill is linked to a specific customer and item because of the foreign key references between those entities and the Bills object.

The database's conceptual design, taken as a whole, offers a high level view of the data and their interconnections. It provides a basis for the logical and physical designs, which concentrate on the database's actual implementation. Developers will be better able to construct a database that serves the needs of the system if they have a firm grasp of its conceptual design.

Below is the ER Diagram for this database application:



Below is the relational Schemas this Database Application:



**Creating the Database Tables**

The following Query was used to create the database tables:

CREATE TABLE Customer (

  Customer\_ID NUMBER(10) PRIMARY KEY,

  Name VARCHAR2(50),

  Address VARCHAR2(100),

  Email VARCHAR2(50),

  Gender VARCHAR2(10)

);

CREATE TABLE Login (

  Acc\_No NUMBER(10) PRIMARY KEY,

  Customer\_ID NUMBER(10) REFERENCES Customer(Customer\_ID),

  Password VARCHAR2(50),

  Actions VARCHAR2(100)

);

CREATE TABLE Product (

  Product\_ID NUMBER(10) PRIMARY KEY,

  ProductName VARCHAR2(50),

  ManufactureDate DATE

);

CREATE TABLE Bills (

  payment\_ID NUMBER(10) PRIMARY KEY,

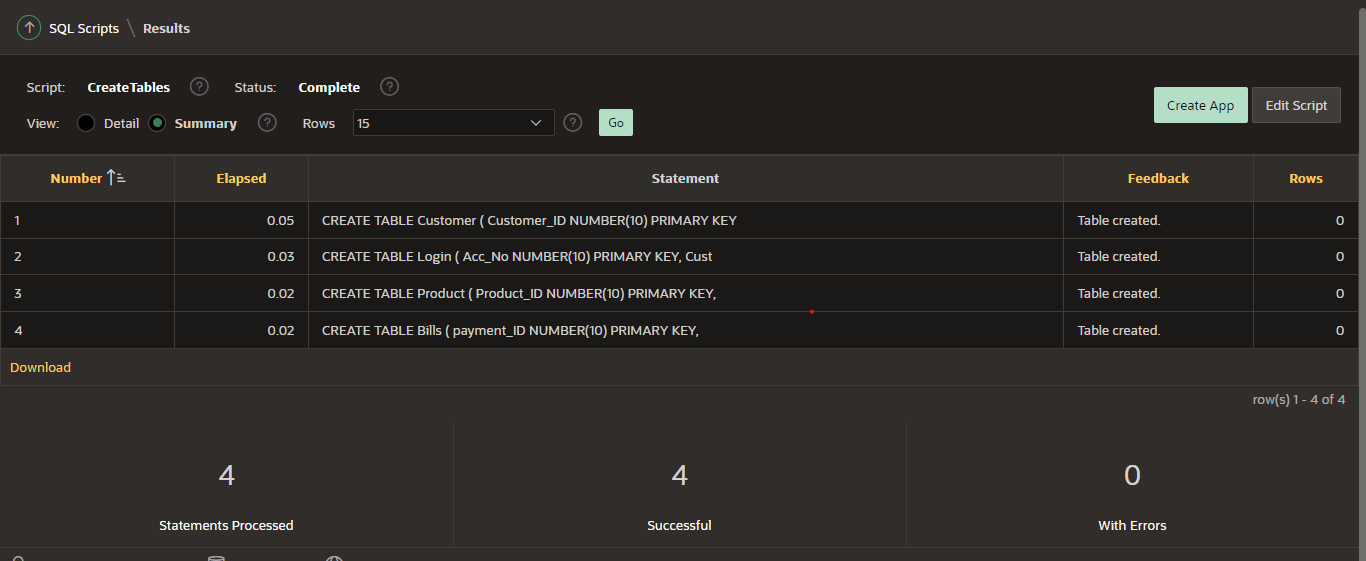
  Customer\_ID NUMBER(10) REFERENCES Customer(Customer\_ID),

  Product\_ID NUMBER(10) REFERENCES Product(Product\_ID),

  PaymentType VARCHAR2(50)

);

**After running the above SQL Code, I got the following results:**

****

**Inserting Data into The Tables**

The following query was used to insert data into the database tables:

INSERT INTO Customer (Customer\_ID, Name, Address, Email, Gender)

VALUES (101, 'John Doe', '123 Main St', 'johndoe@example.com', 'Male');

INSERT INTO Customer (Customer\_ID, Name, Address, Email, Gender)

VALUES (102, 'Jane Smith', '456 Elm St', 'janesmith@example.com', 'Female');

INSERT INTO Customer (Customer\_ID, Name, Address, Email, Gender)

VALUES (103, 'Bob Johnson', '789 Oak St', 'bobjohnson@example.com', 'Male');

INSERT INTO Customer (Customer\_ID, Name, Address, Email, Gender)

VALUES (104, 'Alice Jones', '234 Pine St', 'alicejones@example.com', 'Female');

INSERT INTO Customer (Customer\_ID, Name, Address, Email, Gender)

VALUES (105, 'Mike Davis', '567 Maple St', 'mikedavis@example.com', 'Male');

INSERT INTO Customer (Customer\_ID, Name, Address, Email, Gender)

VALUES (106, 'Samantha Lee', '890 Cedar St', 'samanthalee@example.com', 'Female');

INSERT INTO Login (Acc\_No, Customer\_ID, Password, Actions)

VALUES (1, 101, 'password123', 'view\_profile');

INSERT INTO Login (Acc\_No, Customer\_ID, Password, Actions)

VALUES (2, 102, 'qwerty123', 'make\_payment');

INSERT INTO Login (Acc\_No, Customer\_ID, Password, Actions)

VALUES (3, 103, 'abc123', 'update\_info');

INSERT INTO Login (Acc\_No, Customer\_ID, Password, Actions)

VALUES (4, 104, 'pass123', 'place\_order');

INSERT INTO Login (Acc\_No, Customer\_ID, Password, Actions)

VALUES (5, 105, 'admin123', 'manage\_inventory');

INSERT INTO Login (Acc\_No, Customer\_ID, Password, Actions)

VALUES (6, 106, 'letmein123', 'view\_orders');

INSERT INTO Product (Product\_ID, ProductName, ManufactureDate)

VALUES (201, 'Tesla Model s', TO\_DATE('2022-01-01', 'YYYY-MM-DD'));

INSERT INTO Product (Product\_ID, ProductName, ManufactureDate)

VALUES (202, 'Tesla Model X', TO\_DATE('2022-02-01', 'YYYY-MM-DD'));

INSERT INTO Product (Product\_ID, ProductName, ManufactureDate)

VALUES (203, 'Tesla Battery', TO\_DATE('2022-03-01', 'YYYY-MM-DD'));

INSERT INTO Product (Product\_ID, ProductName, ManufactureDate)

VALUES (204, 'Tesla Charger', TO\_DATE('2022-04-01', 'YYYY-MM-DD'));

INSERT INTO Bills (payment\_ID, Customer\_ID, Product\_ID, PaymentType)

VALUES (1, 101, 201, 'Credit Card');

INSERT INTO Bills (payment\_ID, Customer\_ID, Product\_ID, PaymentType)

VALUES (2, 102, 202, 'Debit Card');

INSERT INTO Bills (payment\_ID, Customer\_ID, Product\_ID, PaymentType)

VALUES (3, 103, 203, 'PayPal');

INSERT INTO Bills (payment\_ID, Customer\_ID, Product\_ID, PaymentType)

VALUES (4, 104, 204, 'Venmo');

INSERT INTO Bills (payment\_ID, Customer\_ID, Product\_ID, PaymentType)

VALUES (5, 105, 205, 'Cash');

INSERT INTO Bills (payment\_ID, Customer\_ID, Product\_ID, PaymentType)

VALUES (6, 106, 206, 'Check');

**Sample Queries**

**Query 1**

**The Query below is used to get the names of customers who purchased a product manufactured after a certain date (e.g. 2022-01-01):**

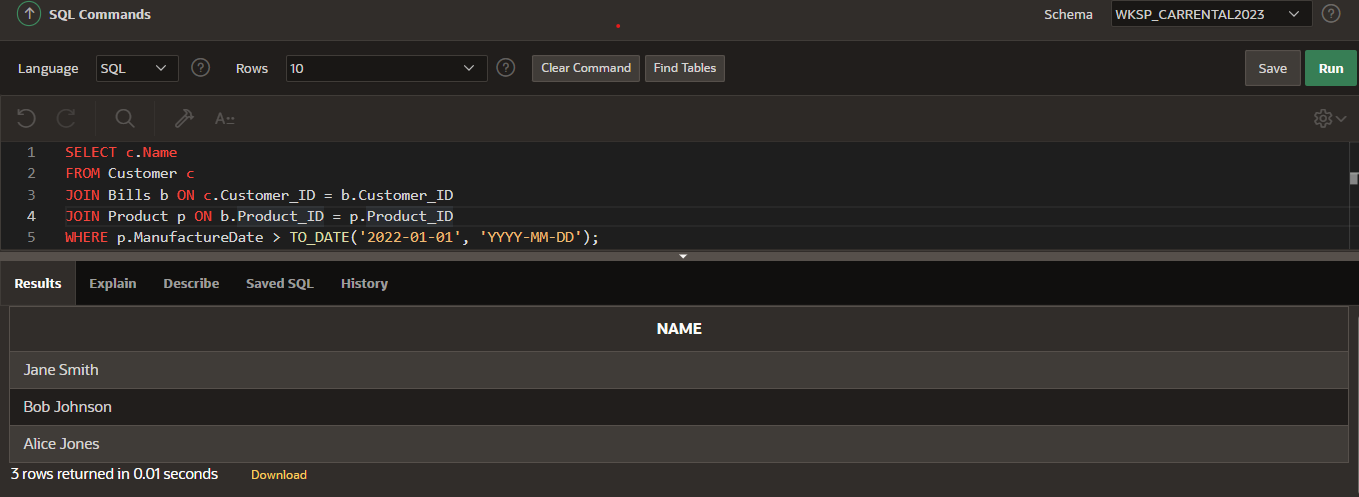
SELECT c.Name FROM Customer c

JOIN Bills b ON c.Customer\_ID = b.Customer\_ID

JOIN Product p ON b.Product\_ID = p.Product\_ID

WHERE p.ManufactureDate > TO\_DATE('2022-01-01', 'YYYY-MM-DD');

**Output:**



**Query 2**

**The Query below gets the product names and total number of sales for each product:**

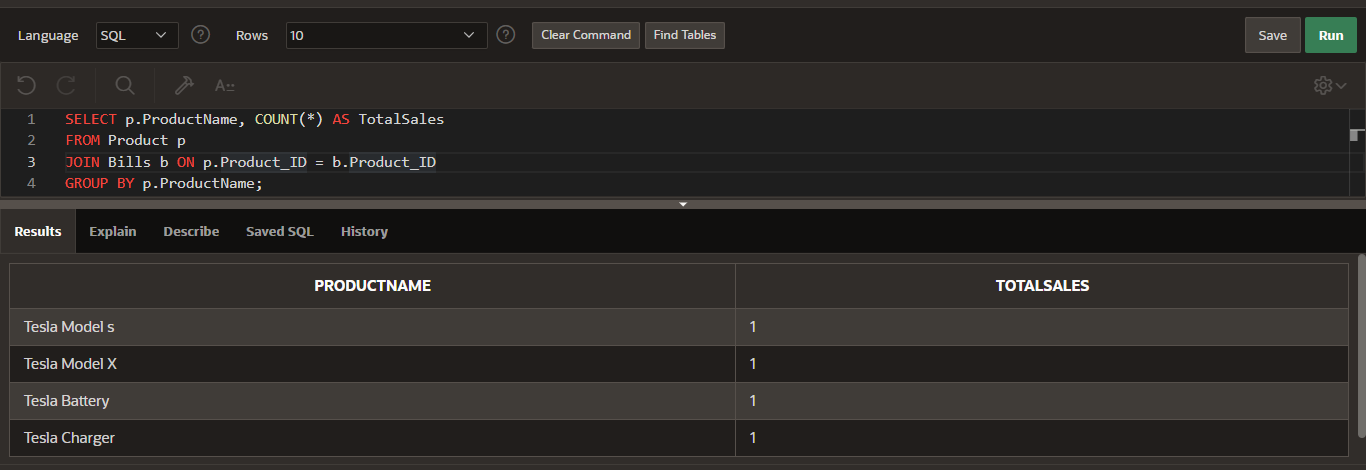
SELECT p.ProductName, COUNT(\*) AS TotalSales

FROM Product p

JOIN Bills b ON p.Product\_ID = b.Product\_ID

GROUP BY p.ProductName;

**Output**

****

**Query 3**

**The query below gets the total number of payments made by each customer:**

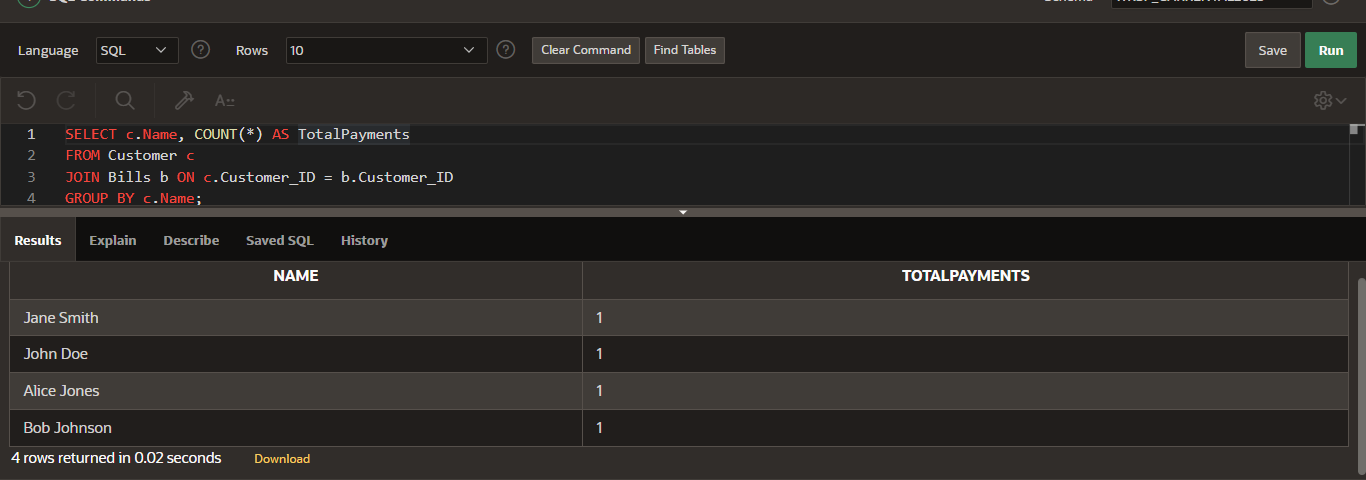
SELECT c.Name, COUNT(\*) AS TotalPayments

FROM Customer c

JOIN Bills b ON c.Customer\_ID = b.Customer\_ID

GROUP BY c.Name;

**Output:**

****

**Query 4**

**The Query below gets the names and email addresses of all customers who made a payment via credit card:**

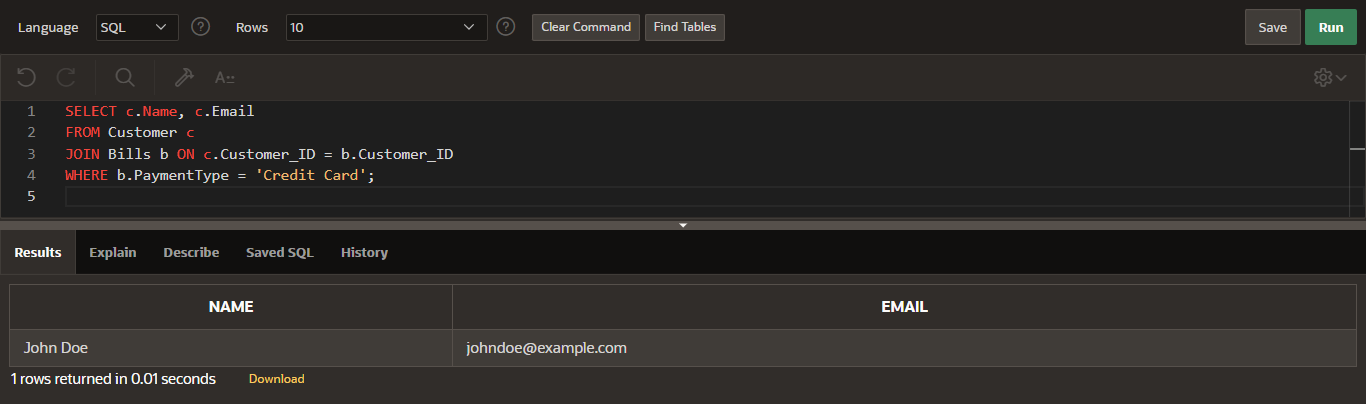
SELECT c.Name, c.Email

FROM Customer c

JOIN Bills b ON c.Customer\_ID = b.Customer\_ID

WHERE b.PaymentType = 'Credit Card';

**Output**

****

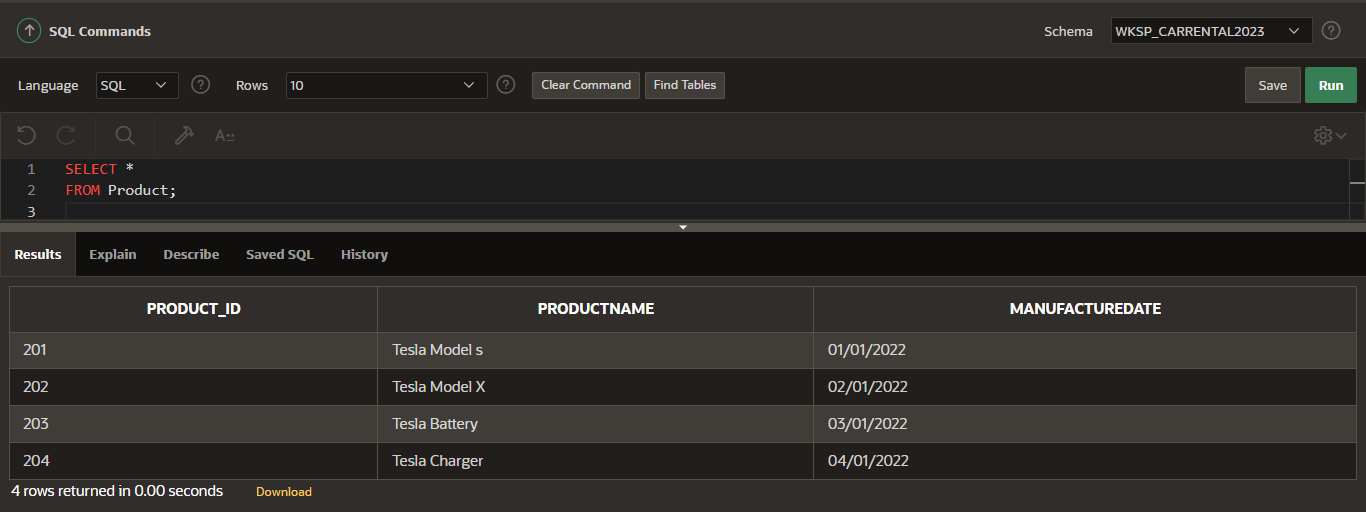
**Query 5**

**The Query below lists all products in the database:**

SELECT \*

FROM Product;

**Output**



**Query 6**

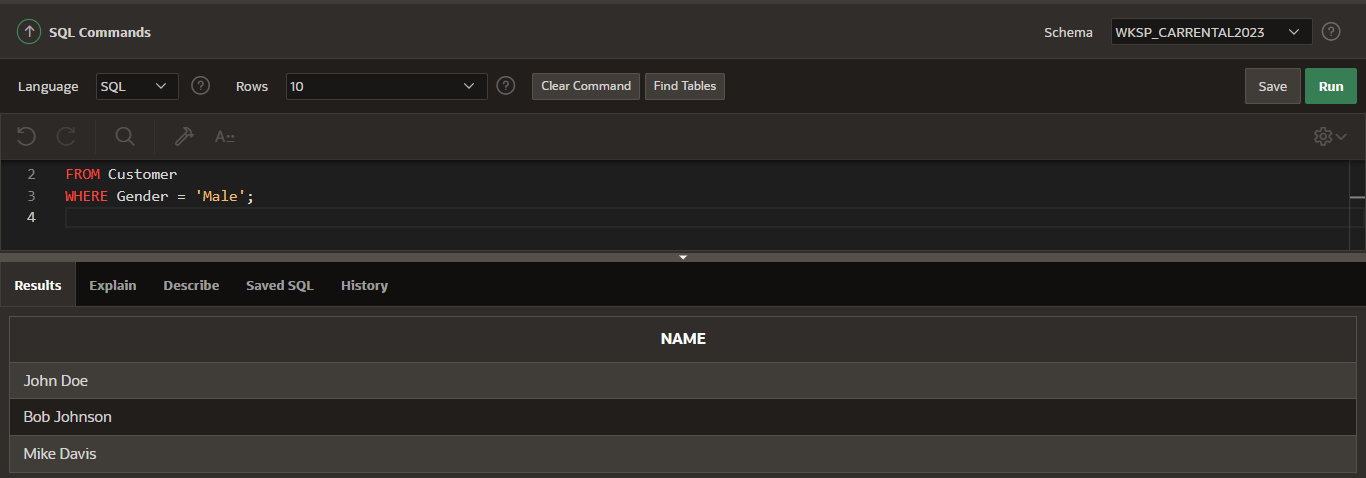
**The query below lists the names of all male customers.**

SELECT Name

FROM Customer

WHERE Gender = 'Male';

**Output**



**Query 7**

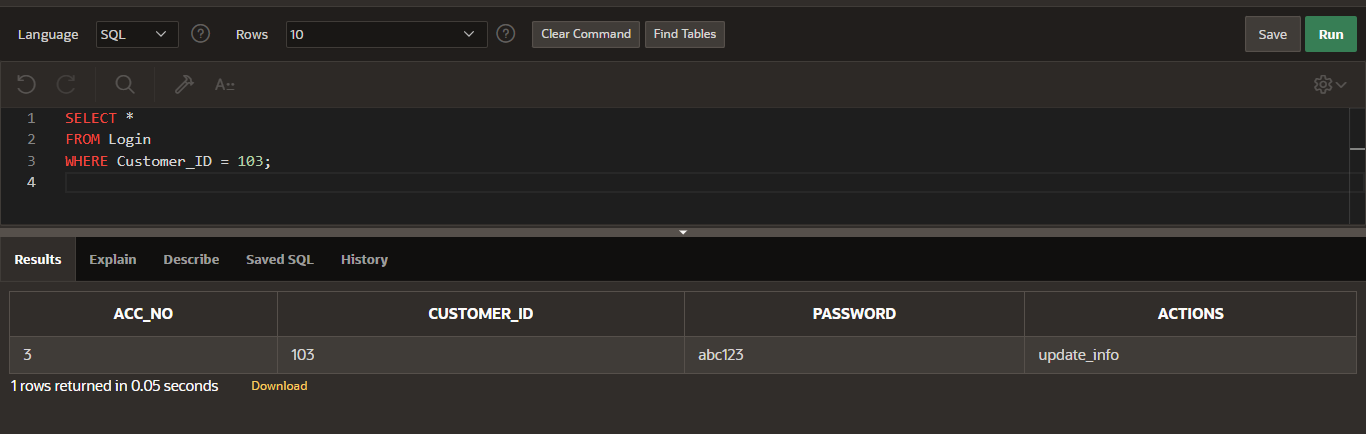
**The query below shows the login details for customer with Customer\_ID = 103**

SELECT \*

FROM Login

WHERE Customer\_ID = 103;

**Output**

****

**Query 8**

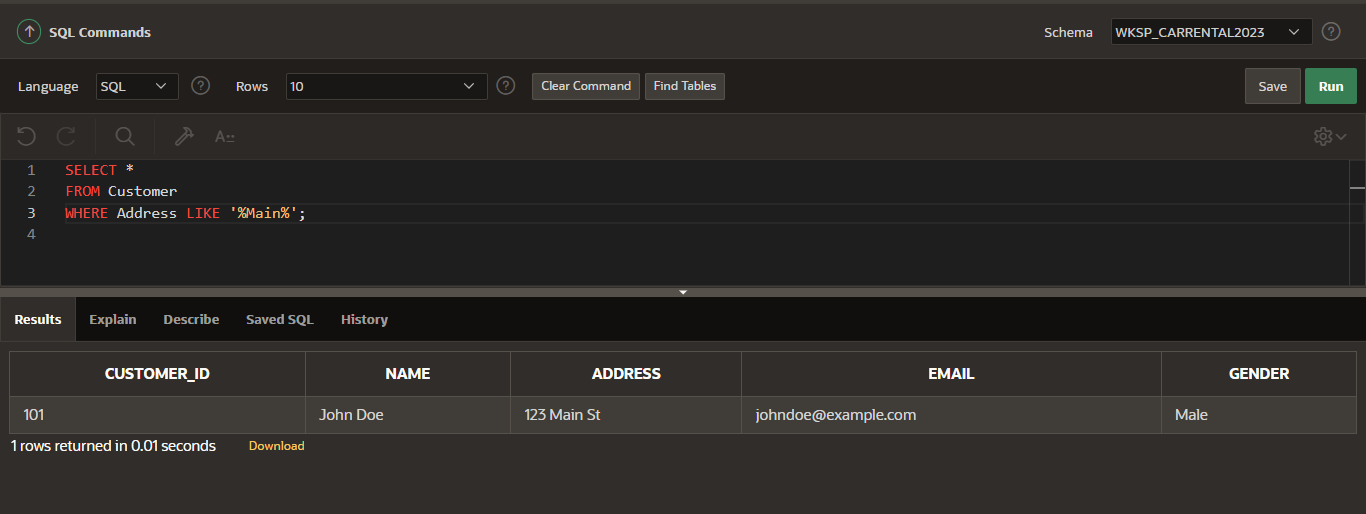
**This query retrieves all customers from the Customer table who have a Main address.**

SELECT \*

FROM Customer

WHERE Address LIKE '%Main%';

**Output**



**References**

Monger, A., Baron, S., & Lu, J. (2009, July). More on Oracle APEX for teaching and learning. In *the 7th International Workshop on Teaching, Learning and Assessment of Databases, 6 July 2009, University of Birmingham* (pp. 3-12).

Nuijten, A., & Barel, P. (2023). Useful APEX Packages. In *Modern Oracle Database Programming: Level Up Your Skill Set to Oracle's Latest and Most Powerful Features in SQL, PL/SQL, and JSON* (pp. 395-418). Berkeley, CA: Apress.

Sciore E. (2020). Understanding oracle apex 20 application development : think like an application express developer (3rd ed.). Apress L.P. https://doi.org/10.1007/978-1-4842-6165-1